



Acceleration Academy 2010

Welcome

Data Analysis Using SQL



New Webinar!
Jan 29, 2010

Guest Speaker -
Gordon Linoff

Register Now for **Acceleration Academy**
First 20 to Register/Attend Get Free SQL Book! >

The complex block is a dark grey rectangular area containing promotional text and an image. At the top, the title 'Data Analysis Using SQL' is written in a light grey font. Below the title is an illustration of a whiteboard on a stand, with a diagram of a circular flow involving 'x' and 'o' characters connected by arrows. To the right of the whiteboard, the text 'New Webinar!' and 'Jan 29, 2010' is displayed. Below that, 'Guest Speaker - Gordon Linoff' is listed. At the bottom of the block, a call to action reads 'Register Now for Acceleration Academy' followed by 'First 20 to Register/Attend Get Free SQL Book!' and a right-pointing arrow.

Please Type in Questions Along the Way

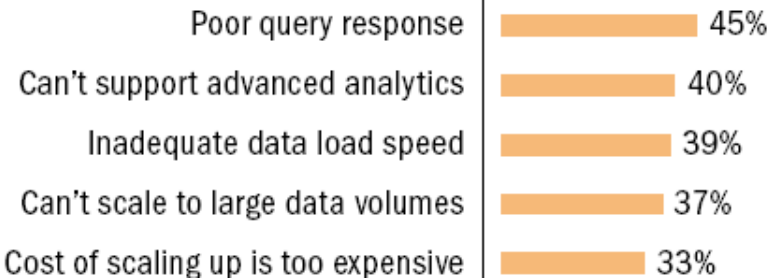
Time for Q/A the end

If we can't get to our question, we will respond
via email

Held last Friday of each month

www.xtremedata.com/accelerationacademy

What problems will eventually drive you to replace your current primary data warehouse platform?
(Select nine or fewer.)



Reference TDWI –
Q4 2009 TDWI Best Practices Report –
Next Generation Data Warehouse Platforms by Philip Russom

http://www.tdwi.org/articles/2009/10/08/tdwi-announces-new-best-practices-report-next-generation-data-warehouse-platforms.aspx?sc_lang=en

- ▶ 4 years in the making
- ▶ Based in Chicago-area
- ▶ Deeply experienced leadership team
- ▶ Our solution
 - ▶ Database appliance - Scalable
 - ▶ Operating range 1TB – 3.8 PB User Data
 - ▶ Ad hoc/unconstrained access
 - ▶ Fast query & load/unload performance
 - ▶ Low TCO
 - ▶ Energy efficient
 - ▶ \$20K/TB Uncompressed – Industry Leading

A graphic consisting of a large red arrow pointing to the right, with a white grid pattern on its tail. The text 'Competing on Analytics' is written in bold black font over the arrow.

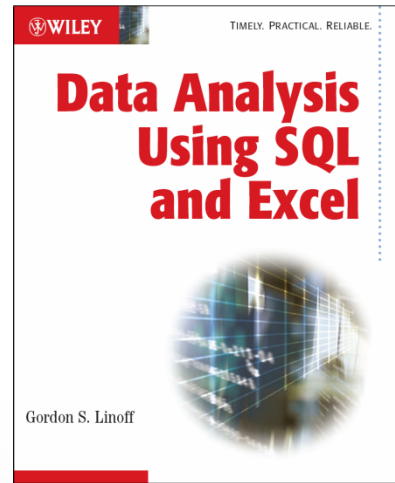
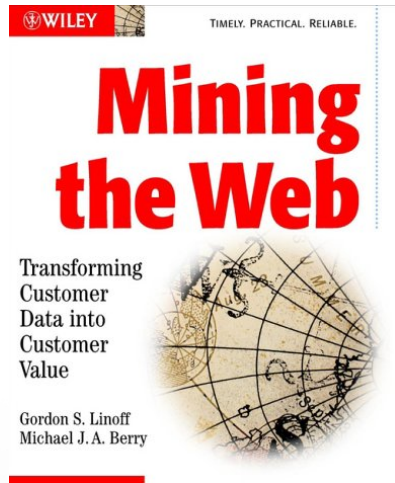
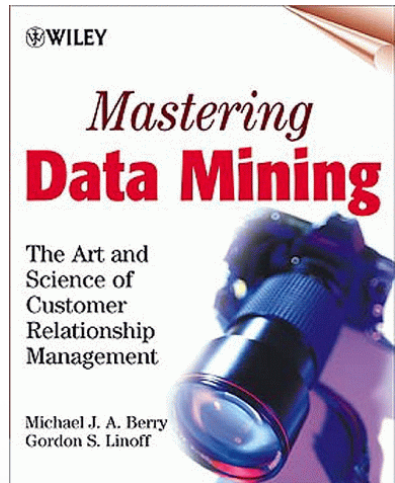
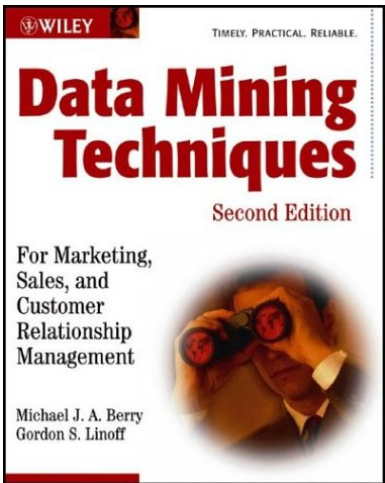
Competing on Analytics

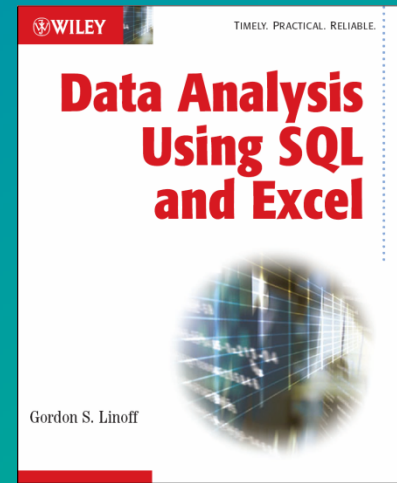
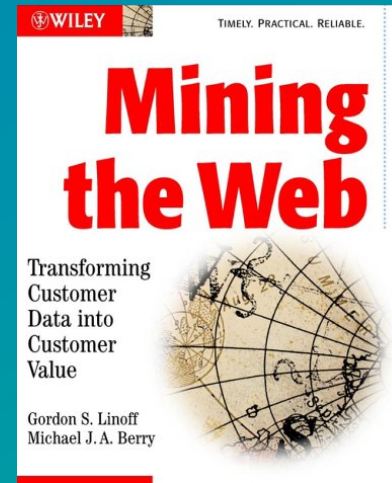
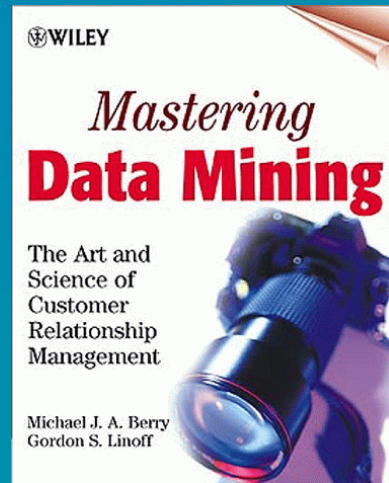
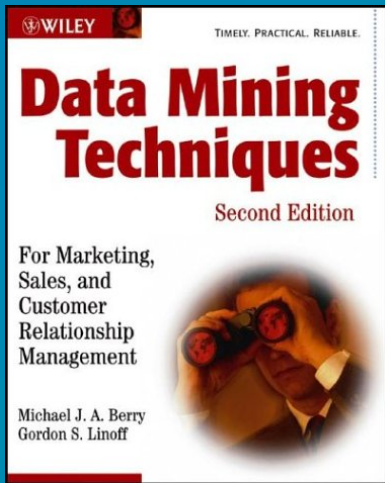
Ad-Hoc Data
Exploration

- “Unrestricted” ad hoc access to data
 - Explore data to unearth patterns, relationships, so forth without having to modify/tune underlying database schema
- Scale to support petabytes of data
 - Deep and wide time series, archived and sub-transactional data
- High performance
 - Compress cycle time for analytics studies - from hypothesis thru validated strategy
- Low cost - energy efficient
 - Hardware/software licensing, support/maintenance, data center, power, cooling
- Easy to use
 - ANSI SQL - standard tools and methods



Gordon Linoff
Principal and Founder of Data Miners
Respected Author and Speaker





Data Analysis Using Relational Databases

Gordon S. Linoff

29 January 2010

Founder

Data Miners, Inc.

gordon@data-miners.com



Agenda

- ◆ My Background
- ◆ Why SQL for data analysis?
- ◆ What parts of SQL are important?
- ◆ Some Examples

Who Am I?

Gordon S. Linoff

- ◆ Author (with Michael Berry) of four books on Data Mining
 - *Data Analysis Using SQL and Excel* (2008)
 - *Data Mining Techniques for Marketing, Sales, and Customer Support, Second Edition* (1998, 2004)
 - *Mastering Data Mining* (2000)
 - *Mining the Web* (2002)
- ◆ Founded Data Miners, Inc. with Michael Berry in 1998
 - Boutique consulting firm focused on data mining
 - Experienced in data warehousing, data mining, parallel computing, parallel databases
 - Teach classes on data mining (primarily through SAS Institute)
- ◆ Worked at Thinking Machines in the early 1990s developing a parallel database engine for the Connection Machine

Data Mining

- ◆ Hypothesis Testing and Measurement
 - Understanding the data and whether or not our hunches are correct
- ◆ Predictive Modeling and Profiling
 - Building models to predict some desired outcome or to understand the effects of inputs on the outcome
- ◆ Undirected Data Mining
 - Looking for what you don't know in the data

Data Mining usually focuses on sexy algorithms, like neural networks and decision trees. Hypothesis testing and measurement are just as important, and very well suited to RMDBS.

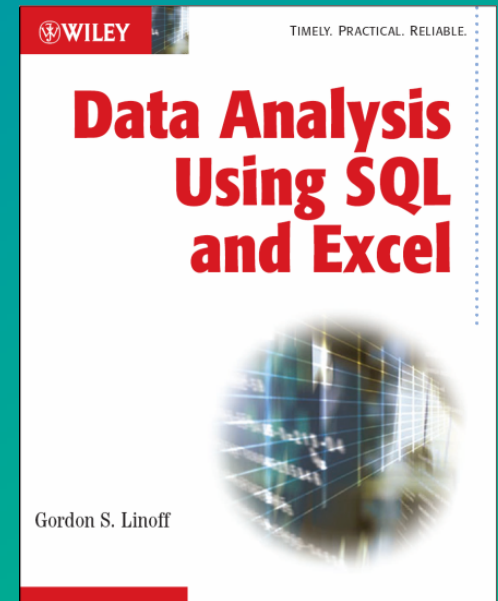
The Real Story

So what are the facts? In 1992, Thomas Blischok, manager of a retail consulting group at Teradata, and his staff prepared an analysis of 1.2 million market baskets from about 25 Osco Drug stores. Database queries were developed to identify affinities. The analysis "did discover that between 5:00 and 7:00 p.m. that consumers bought beer and diapers". Osco managers did NOT exploit the beer and diapers relationship by moving the products closer together on the shelves. This decision support study was conducted using query tools to find an association. The true story is very bland compared to the legend.

*Daniel Power, at
<http://dssresources.com/newsletters/66.php>*

There Is A Lot You Can Do With SQL

- ◆ A Data Miner Looks At SQL
- ◆ What's In A Table: Getting Started With Data Exploration
- ◆ How Different Is Different?
- ◆ Where Is It All Happening? Location, Location, Location
- ◆ It's A Matter of Time
- ◆ How Long Will Customers Last? Survival Analysis to Understand Customers and Their Value
- ◆ Factors Affecting Survival: The What and Why of Customer Tenure
- ◆ Customer Purchases and Other Repeated Events
- ◆ What's in a Shopping Cart? Market Basket Analysis and Association Rules
- ◆ Data Mining Models in SQL
- ◆ The Best Fit Line: Linear Regression Models
- ◆ Building Customer Signatures for Further Analysis



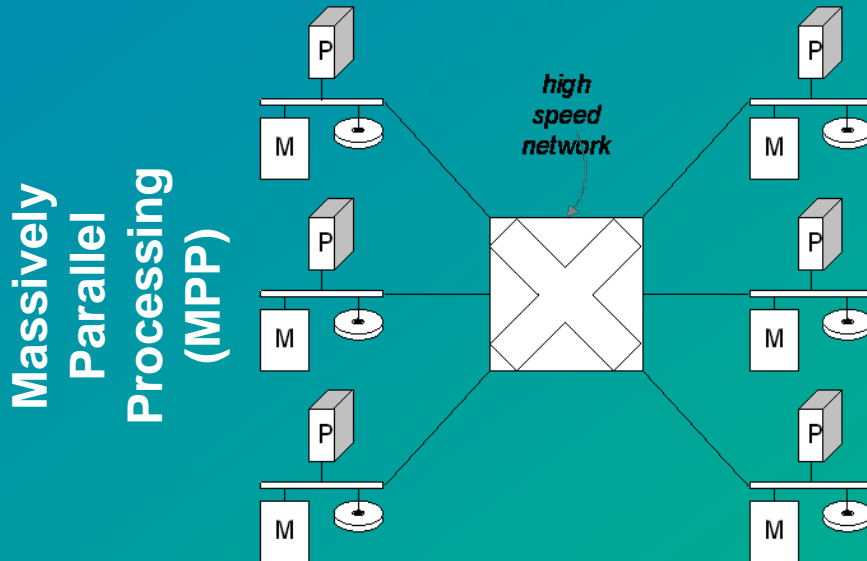
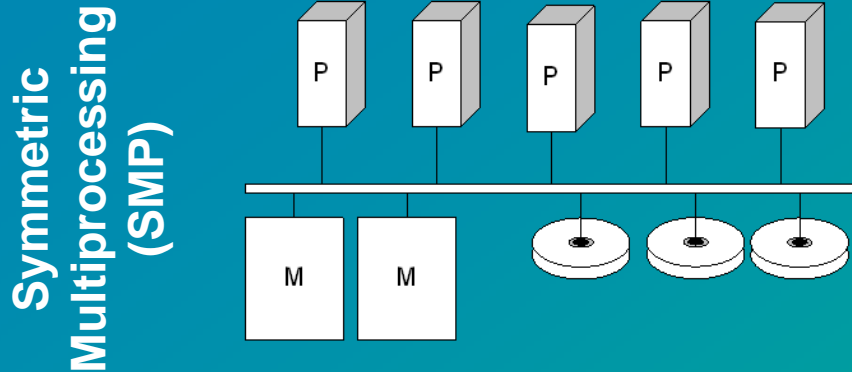
And I'm very proud of my 10/10 Five Star Reviews on Amazon!

WHY IS SQL USEFUL FOR DATA ANALYSIS?

Why Is SQL Useful for Data Analysis?

- ◆ Relational databases have been the only systems available that take advantage of parallelism (multiple disks, multiple processors, lots of memory)
- ◆ There are numerous products available from a wide variety of vendors
 - And since this talk is sponsored by one of them, I won't attempt to list them
- ◆ SQL is a common language with standards
- ◆ You can hire people who know SQL
- ◆ SQL is natural for creating and presenting reporting systems
- ◆ Our analytic problem is often finding anything of interest in zillions of bytes of data
- ◆ And yet, with SQL we can implement some sophisticated statistics pretty easily

The Power of Parallel Computing



- ◆ Parallel relational database takes advantage of multiple disks, multiple processors, lots of memory
- ◆ Parallel relational databases support SQL
- ◆ MPP is superior to SMP for scalability

What Are the Options for Big Problems on Parallel Machines?

- ◆ Relational databases
 - Many products from many vendors
 - Standards
 - Proven technology around since the 1980s
- ◆ MapReduce/Hadoop and the like
 - Freeware
 - Requires programming
 - Lots of fun for nerds, perhaps less fun for business people
- ◆ ETL Tools
 - Lots of data analysis is very fancy ETL
 - Ab Initio and Informatica are very expensive
- ◆ Wait until the computers get more powerful
 - Not so unrealistic, given Moore's Law
 - But, we have problems to solve today

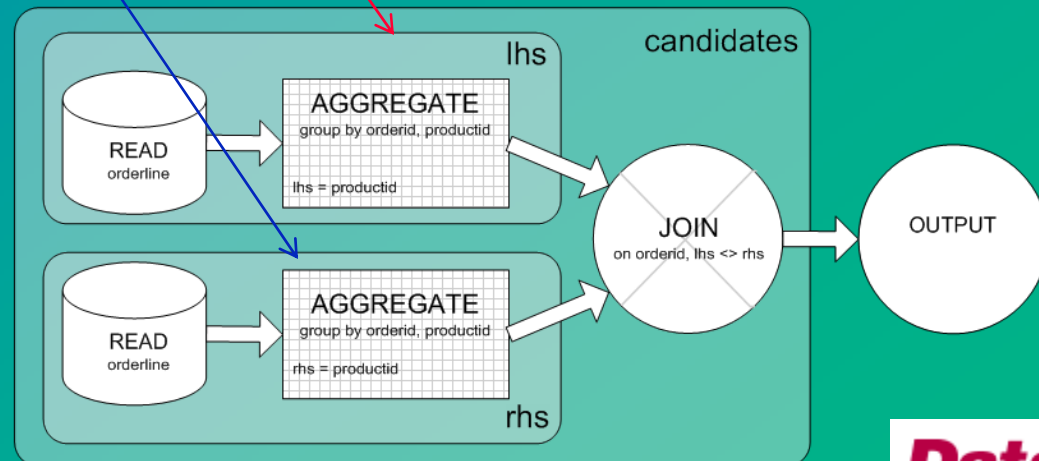
Advantages of SQL for Data Analysis

- ◆ Standard language
 - Sort of
- ◆ Lots of people learn SQL
- ◆ SQL is advantageous for building reports and interfacing to other systems
- ◆ Declarative not programmatic
 - Describe what the results look like rather than how to get them
 - Perhaps better in theory than in practice for large, complex queries
- ◆ Hides almost all complexity related to parallel hardware
 - Queries look the same regardless of the architecture of the machine running them
 - The optimizer does all the work

SQL Implements Dataflows: Finding Pairs of Products in the Same Order

```
SELECT lhs.orderid, lhs.lhs, rhs.rhs
FROM (SELECT DISTINCT orderid, productid as lhs
      FROM orderline
    ) lhs JOIN
      (SELECT DISTINCT orderid, productid as rhs
      FROM orderline) rhs
ON lhs.orderid = rhs.orderid AND
   lhs.lhs <> rhs.rhs
```

This query calculates all candidate rules for two-way associations. These are rules of the form:
<lhs> → <rhs>



WHAT PARTS OF THE LANGUAGE ARE IMPORTANT FOR DATA ANALYSIS?

High Level View of What's Important

USED

- ◆ **SELECT**
- ◆ **SELECT**
- ◆ **SELECT**
- ◆ Sometimes “**CREATE TABLE AS**” or “**INSERT INTO**”

NOT USED

- ◆ **UPDATE**
- ◆ **DELETE**
- ◆ **INSERT**
- ◆ And most of the other things mentioned in the standard

More Specifics On What Gets Used

- ◆ Subqueries
 - But not correlated subqueries (almost never needed and generally introduce serialization)
- ◆ GROUP BY
 - COUNT DISTINCT
 - GROUPING SETS are nice to have
- ◆ Inner joins, Left Outer Joins, Full Outer Joins
- ◆ Window Functions
 - E.g., ROW_NUMBER() OVER (PARTITION BY xxx ORDER BY yyy)
- ◆ Good text processing and date and time functions
- ◆ Efficient User Defined Functions
- ◆ Indexes are less important than efficient parallel join and aggregation operations

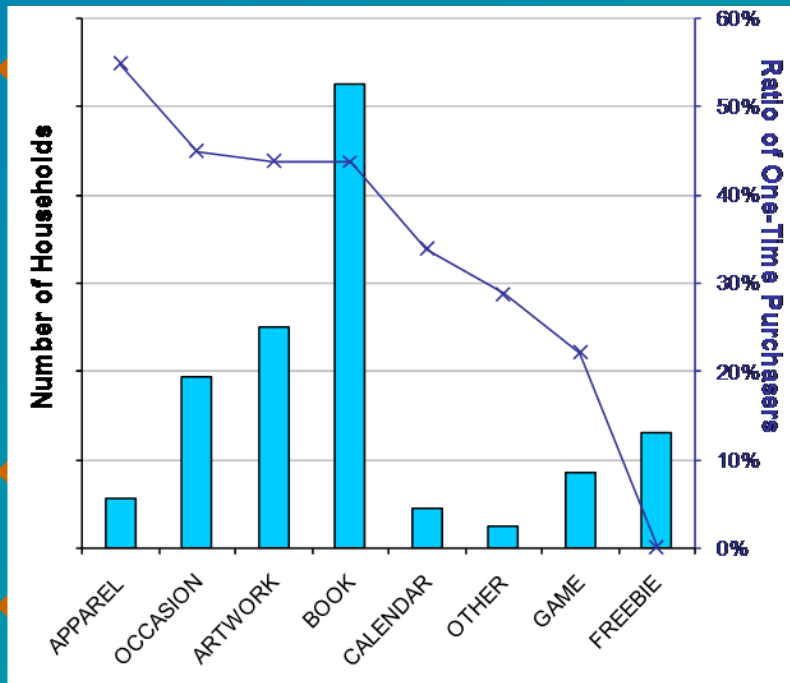
Lots of Big, Complicated Queries

- ◆ In some circumstances, it is not possible or desirable to write to the database.
- ◆ Explicitly storing results in intermediate tables might prevent the optimizer from choosing the most efficient query plan.
- ◆ Doing an analytic task in one query (versus a script file) saves time and effort in storage management.
- ◆ Using subqueries eliminates any dependencies on possibly inefficient temporary user storage (such as lack of partitioning).
- ◆ Subqueries allow the optimizer to eliminate unused columns from intermediate tables.

(from my blog post <http://www.data-miners.com/blog/2008/07/nested-subqueries-in-sql.html>)

SOME EXAMPLES

Which Products Are Associated with One-Time Purchasers?



Actionable information: suggests customer segments (based on initial purchase category) and product groups

The SQL Is Perhaps More Complicated Than We Would Expect

What is the ratio of “unique” purchasers for each product group?
What is the ratio of “unique” purchasers for each product?

How many households purchase each product?

How many such one-product households does each product have?
Which households have exactly one order and one product?

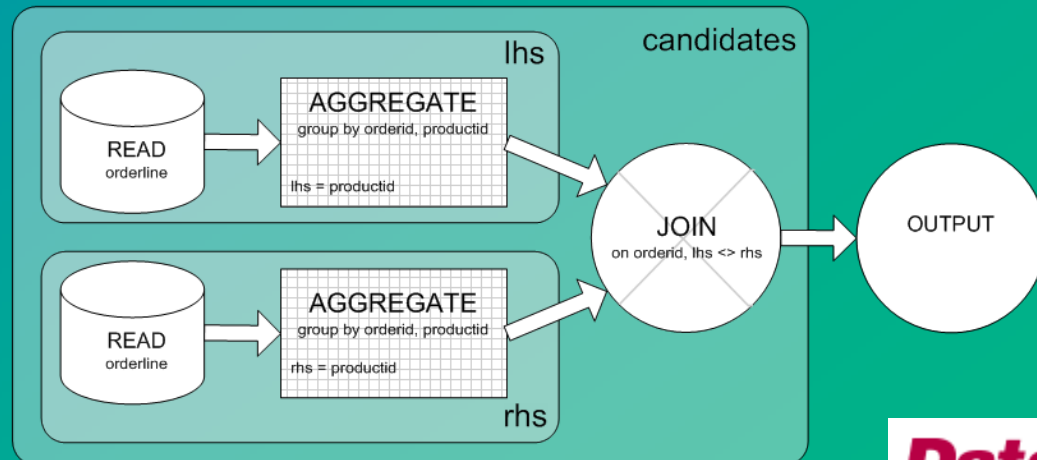
```
SELECT productgroupname, COUNT(*) as numprods,
      SUM(numhouseholds) as numhh,
      SUM(numuniques) as numuniques,
      SUM(numuniques*1.0)/SUM(numhouseholds) as ratio
FROM (SELECT p.productid, numhouseholds, numuniques,
      numuniques*1.0 / numhouseholds as prodratio
FROM (SELECT productid, COUNT(*) as numhouseholds
FROM (SELECT c.householdid, ol.productid
FROM customer c JOIN
      orders o
      ON c.customerid = o.customerid JOIN
      orderline ol
      ON o.orderid = ol.orderid
GROUP BY c.householdid, ol.productid
) hp
GROUP BY productid) p LEFT OUTER JOIN
(SELECT productid, COUNT(*) as numuniques
FROM (SELECT householdid,
      MIN(productid) as productid
FROM customer c JOIN
      orders o
      ON c.customerid = o.customerid JOIN
      orderline ol ON o.orderid = ol.orderid
GROUP BY householdid
HAVING COUNT(DISTINCT ol.productid) = 1 AND
      COUNT(DISTINCT o.orderid) = 1) h
GROUP BY productid
) hp
ON hp.productid = p.productid) hp JOIN
      product p ON hp.productid = p.productid
GROUP BY productgroupname
ORDER BY 5 DESC
```

What Is An Association Rule?

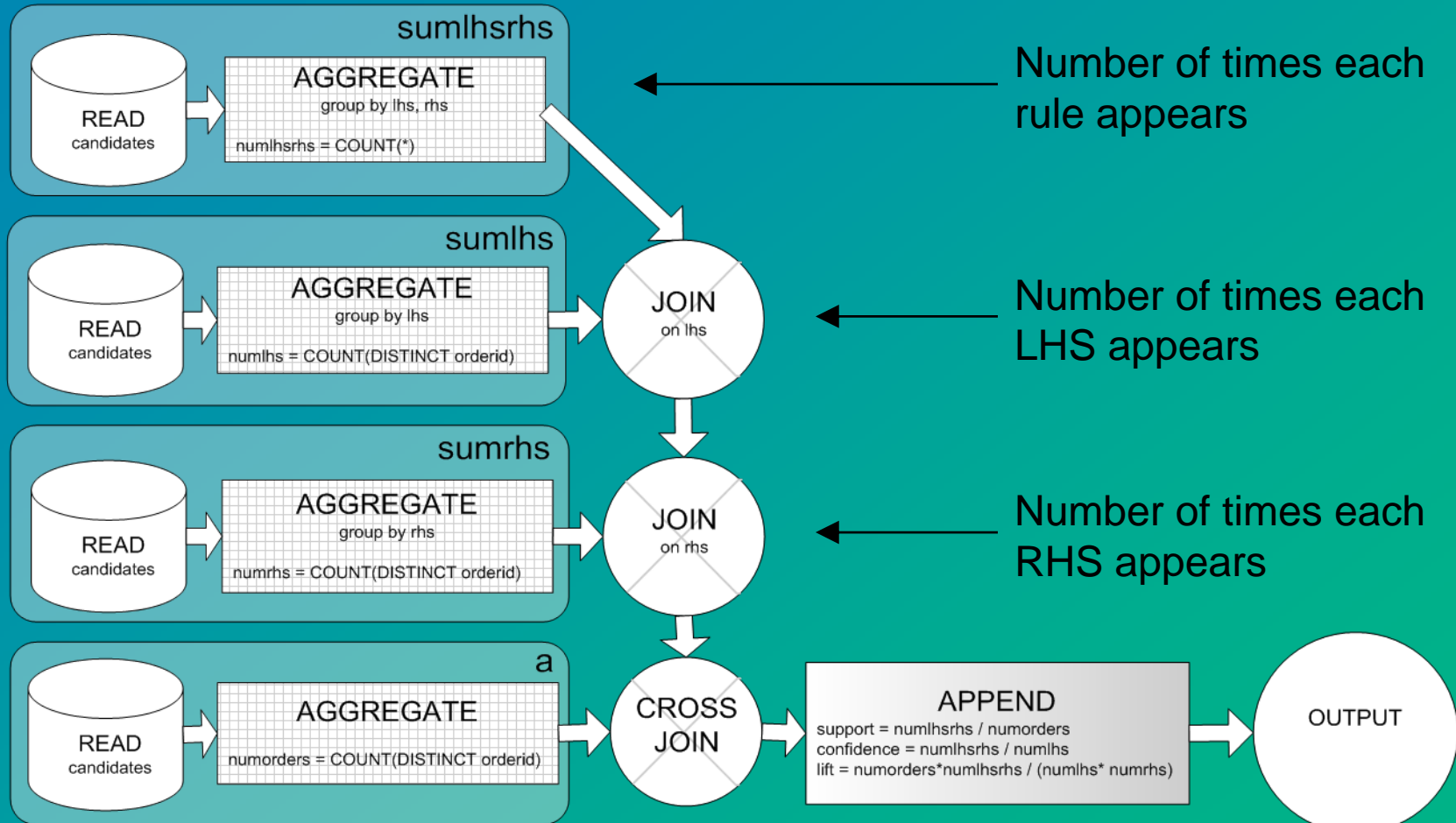
- ◆ Associations rules tell us what products or events happen to occur together
- ◆ When products simply tend to occur together, we call that an *item set*
- ◆ LHS → RHS
 - “When the products on the left hand side are present in a transaction, then the products on the right hand side are present”
 - LHS = left hand side. It typically consists of zero or more products
 - RHS = right hand side. It typically consists of one product
- ◆ Make recommendations in online e-tailing
- ◆ Segment web customers by the areas on the site that they visit
- ◆ Recommend ring tones
- ◆ Cross-sell financial products
- ◆ Clean data – exceptions to very common rules suggest data issues

Generating Rules: Start with the Candidates

```
SELECT lhs.orderid, lhs.lhs, rhs.rhs
FROM (SELECT DISTINCT orderid, productid as lhs
      FROM orderline
    ) lhs JOIN
      (SELECT DISTINCT orderid, productid as rhs
      FROM orderline) rhs
ON lhs.orderid = rhs.orderid AND
   lhs.lhs <> rhs.rhs
```



Measure The Goodness By Analyzing Counts



Looking at the Calculation in SQL

```
SELECT (SQUARE(explhsrhs - numlhsrhs)/explhsrhs +
        SQUARE(explhsnorhs - numlhsnorhs)/explhsnorhs +
        SQUARE(expnolhsrhs - numnolhsrhs)/expnolhsrhs +
        SQUARE(expnolhsnorhs - numnolhsnorhs)/expnolhsnorhs) as chisquare,
        b.*
FROM (SELECT lhsrhs.*, numorders, numlhs, numrhs,
            numlhs - numlhsrhs as numlhsnorhs,
            numrhs - numlhsrhs as numnolhsrhs,
            numlhs*numrhs/numorders as explhsrhs,
            numlhs*(numorders-numrhs)/numorders as explhsnorhs,
            (numorders-numlhs)*numrhs/numorders as expnolhsrhs,
            (numorders-numlhs)*(numorders-numrhs)/numorders) as expnolhsnor,
            (numorders - numlhs - numrhs + numlhsrhs) as numnolhsnorhs,
            numlhsrhs/numorders as support,
            numlhsrhs/numlhs as confidence,
            numlhsrhs * numorders/(numlhs * numrhs) as lift
FROM (<LHS→RHS subquery> ) sumlhsrhs JOIN
     (<LHS subquery>) sumlhs
ON lhsrhs.lhs = sumlhs.lhs JOIN
     (<RHS subquery>) sumrhs
ON lhsrhs.rhs = sumrhs.rhs CROSS JOIN
     (<ALL subquery>) a
```

Association Rules Are Not Always Interesting

Rule	Support	Lift	Chi-Square
NewsNationalAny + OpinionEdOpEdAny ==> NewsWeatherAny	0.75%	1.75	137,622,787.2
NewsNationalAny + NewsPoliticsAny ==> NewsWeatherAny	0.73%	1.74	136,576,999.4
NewsNationalAny + NewsSportsAny ==> NewsWeatherAny	0.67%	2.24	115,246,012.2
NewsNationalAny + NewsTechAny ==> NewsWeatherAny	0.57%	1.98	113,598,808.4
NewsBusinessAny + NewsNationalAny ==> NewsWeatherAny	0.77%	1.81	111,732,631.4
NewsNationalAny + NewsScienceAny ==> NewsWeatherAny	0.53%	1.95	111,683,229.3
NewsInternationalAny + NewsNationalAny ==> NewsWeatherAny	0.78%	1.82	107,702,669.5
NewsBusinessAny + NewsInternationalAny ==> NewsWeatherAny	0.73%	1.88	105,603,103.1
NewsBusinessAny + OpinionEdOpEdAny ==> NewsWeatherAny	0.69%	1.84	104,449,101.8
NewsBusinessAny + NewsHealthAny ==> NewsWeatherAny	0.68%	1.90	101,293,242.1

Hmmm . . . Weather is a popular part of the site among customers that visit multiple parts of the site

Why Do We Want to do Association Rules in SQL?

- ◆ Tools that implement association rules sometimes have limits on the number of rules considered.
- ◆ The Chi-Square measure is the best measure for association rules, but tools do not support it.
- ◆ SQL can support variations on association rules, such as:
 - Purchases made by a household, but not at the same time.
 - Purchases made by a household, in a particular order (sequential rules).
 - Purchases made by a household within a particular span of time.
 - Mixing customer attributes and product attributes in the mix.
- ◆ We may want heterogeneous examples, say, where the left and right and sides are different.

Example of Heterogeneous Rules: Click + Click → Complaint

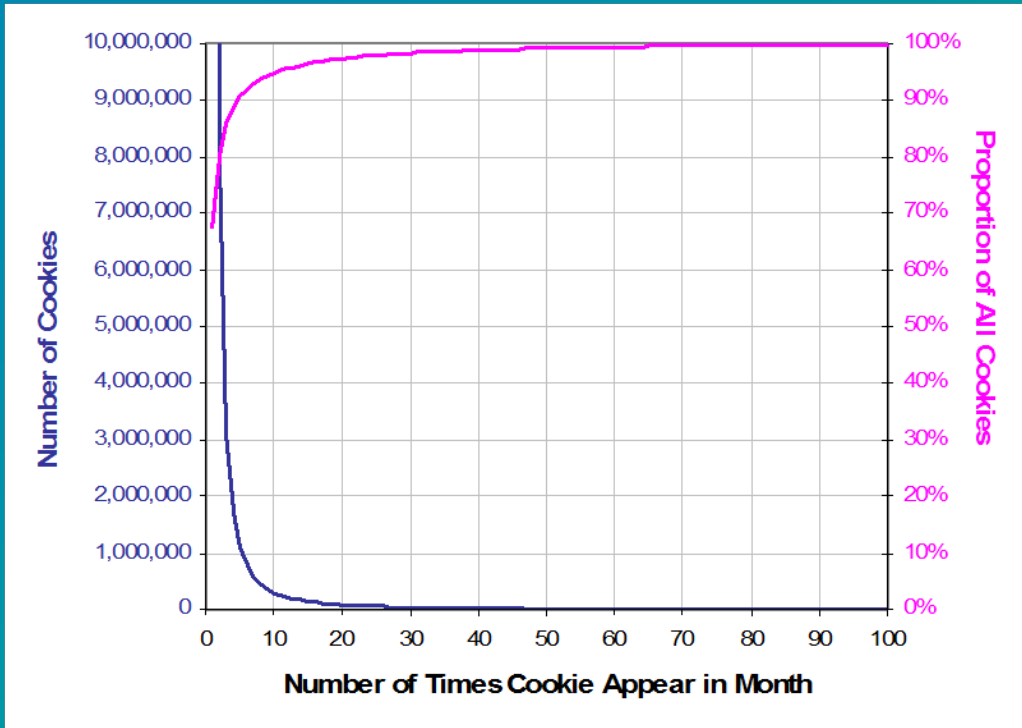
Clicks Imply Complaint Rules	Chi-Square
Telecom + Travel ==> Loans	299.0
Telecom + Government Grants ==> Credit Report	299.0
Government Grants + Gifts ==> Credit Report	299.0
Education + College/Scholarship ==> [Uncategorized]	149.0
Debt + Telecom ==> Credit Report	149.0
Debt + Government Grants ==> Credit Report	149.0
Debt + Gifts ==> Credit Report	149.0
Credit Card + Travel ==> Loans	99.0
Credit Card + Government Grants ==> Credit Report	99.0
Entrepreneurial + Credit Report ==> Home Improvement	74.0

- (1) Customers do not like receiving email offers about Credit Reports.
- (2) They also do not like offers radically different from their past interests

How Many Visitors Come to a Web Site in One Month?

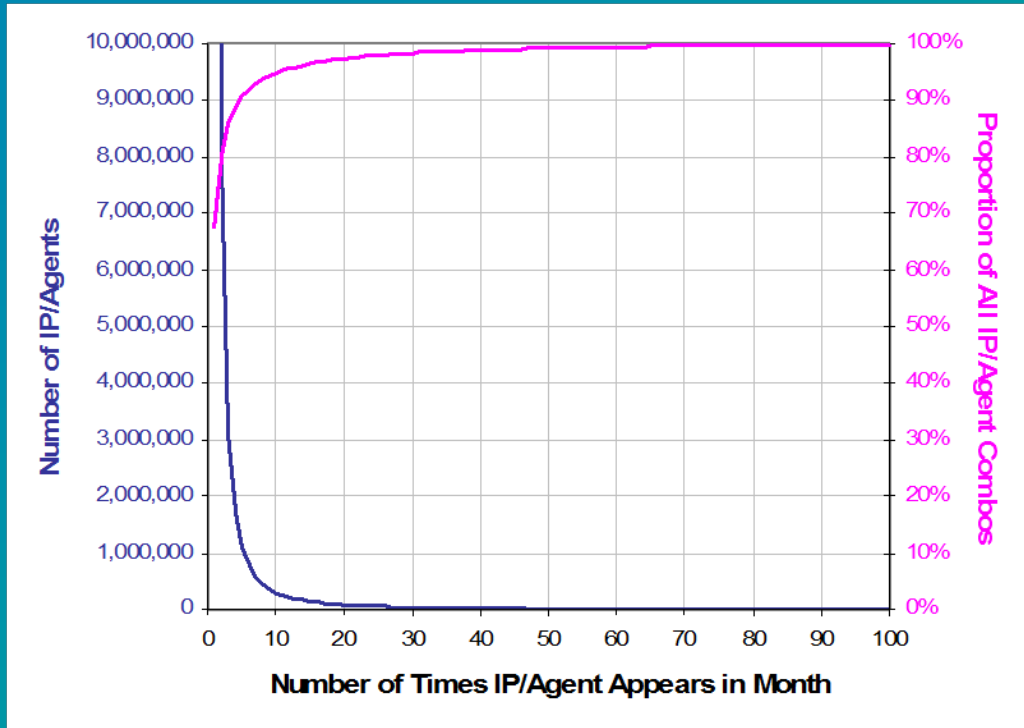
- ◆ Question asked by a major media web site
 - Over 1,000,000,000 page hits per month
 - Over 200,000,000 sessions per month
 - Tens of millions of unique cookies seen on web site
 - Several terabytes of data
- ◆ Some small proportion are registered
- ◆ Number of Users is a Key business metric
- ◆ Data has cookies, ip addresses, agent, registration ids
- ◆ Work done using Xtremedata database

Counting Cookies in One Month



- ◆ Most cookies appear a handful of times (65% appear only once).
- ◆ Some appear more than 100 times (0.3%)
 - Unlikely to be a single user
- ◆ L-shaped distribution

Counting IP Address/Agent Combinations in One Month



- ◆ Most cookies appear a handful of times (68% appear only once).
- ◆ Some appear more than 100 times (0.2%)
 - Unlikely to be a single user
- ◆ L-shaped distribution

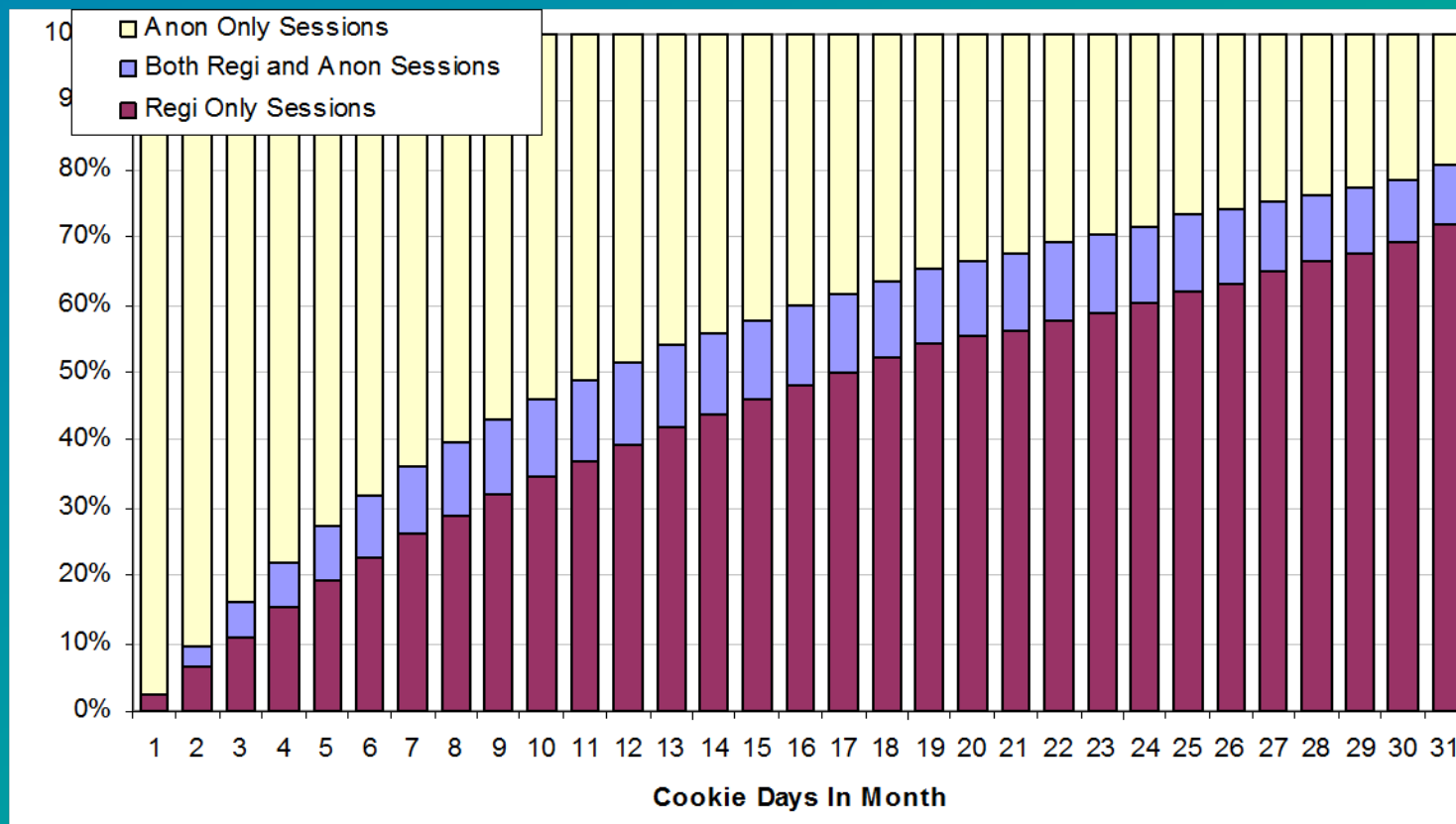
A Typical Histogram of Histogram Query

```
SELECT yr, mon,
       (CASE WHEN numcookies > 100 THEN 101
            ELSE numcookies END) as numcookies,
       COUNT(*),
FROM (SELECT EXTRACT(year FROM start_date) as yr,
            EXTRACT(month FROM start_date) as mon,
            cookie_id, count(*) as numcookies
      FROM (SELECT ps.*,
                  DATE '1970-01-01' +
CAST(start_time/(24*60*60) as INTEGER) as start_date
            FROM psf_session ps
          ) ps
      WHERE cookie_id <> 0
      GROUP BY 1, 2, 3
     ) a
GROUP BY 1, 2, 3
ORDER BY 1, 2, COUNT(*) desc
```

Users Versus Cookies and IP/Addr Combos

- ◆ Users use more than one machine and more than one browser
 - Each of these result in multiple cookies
- ◆ Users may use common machines (such as in airport lounges)
 - Each of these result in multiple cookies for one user
- ◆ Networks may have only one ip address point of presence on the internet
 - One IP address/Agent combo for zillions of users
 - Historically a problem for AOL
- ◆ Networks may not allow cookies
- ◆ We need to use logins as guidance

Are frequent users more likely to be registered?



Are common users more likely to be registered?

```
SELECT yyyyymm, cookiedayspermonth, count(*), SUM(haslogin) as cookiesonlogins,
      SUM(hasnologin) as cookiesonnologins, SUM(haslogin*hasnologin) as cookiesonboth,
      SUM(haslogin*(1-hasnologin)) as cookiesonloginonly,
      SUM((1-haslogin)*hasnologin) as cookiesonnologinonly
FROM (SELECT c.cookie_id, c.yyyyymm, COUNT(distinct thedate) as cookiedayspermonth,
      MAX(haslogin) as haslogin, MAX(hasnologin) as hasnologin
      FROM (SELECT ps.*,
            (DATE '1970-01-01'+CAST(start_time/(24*60*60) as INTEGER)) as thedate,
            TO_CHAR(date '1970-01-01' + CAST(start_time/(24*60*60) as INTEGER),
            'YYYY-MM') as yyyyymm
            FROM public.psf_session ps
            ) c LEFT OUTER JOIN
            (SELECT ps.cookie_id,
            TO_CHAR(DATE '1970-01-01' + CAST(start_time/(24*60*60) as INTEGER),
            'YYYY-MM') as yyyyymm,
            MAX(CASE WHEN login_id > 0 THEN 1 ELSE 0 END) as haslogin,
            MAX(CASE WHEN login_id = 0 THEN 1 ELSE 0 END) as hasnologin
            FROM public.psf_session ps
            GROUP BY 1, 2
            ) cl
            ON c.cookie_id = cl.cookie_id AND c.yyyyymm = cl.yyyyymm
      GROUP BY 1, 2
      ) 1
GROUP BY 1, 2
ORDER BY 1, 2
```

Counting Users

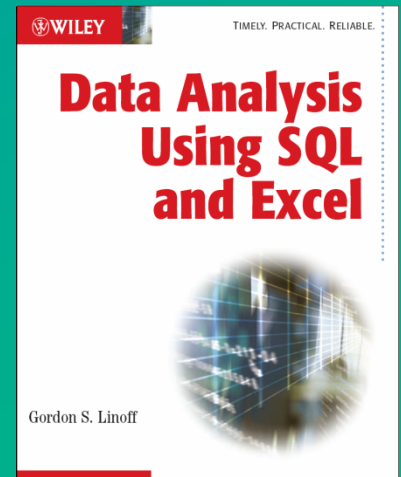
- ◆ Used registered sessions as the standard
- ◆ Calculate average number of registered users per unique cookie on registered session
- ◆ Multiply by total number of unique cookies
- ◆ Get the total number of users

The Final Calculation Stratified the User Sessions

USAGE	ENTER	LOGINS	REF NONE	REF EXT	REF INT	Logins/ Session	Prop of Cookies	Prop of Sessions	Prop of Logins
ONEPAGE	NOHOME	NOLOGIN	0	1	0	0.7665	16.5%	3.56%	19.3%
ONESESSION	NOHOME	NOLOGIN	0	1	0	0.6962	14.7%	3.18%	15.6%
ONESESSION	ALLHOME	NOLOGIN	1	0	0	0.4610	10.1%	2.17%	7.1%
MULTIPLE	NOHOME	NOLOGIN	0	1	0	0.7674	8.6%	5.88%	10.0%
ONEPAGE	NOHOME	NOLOGIN	1	0	0	0.6950	7.7%	1.67%	8.2%
MULTIPLE	ALLHOME	NOLOGIN	1	0	0	0.5343	6.1%	10.20%	5.0%
ONESESSION	NOHOME	NOLOGIN	1	0	0	0.5592	5.1%	1.10%	4.4%
MULTIPLE	SOMEHOME	NOLOGIN	1	1	0	0.6672	2.8%	5.15%	2.8%
MULTIPLE	NOHOME	NOLOGIN	1	1	0	0.7709	2.4%	2.12%	2.8%
MULTIPLE	NOHOME	NOLOGIN	1	0	0	0.7138	2.1%	2.11%	2.3%
ONEPAGE	ALLHOME	NOLOGIN	1	0	0	0.5670	1.7%	0.37%	1.5%
ONESESSION	ALLHOME	NOLOGIN	0	1	0	0.6066	1.7%	0.36%	1.6%
ONESESSION	NOHOME	NOLOGIN	0	0	1	0.5754	1.6%	0.35%	1.4%
MULTIPLE	ALLHOME	ALLLOGIN	1	0	0	0.5343	1.4%	5.20%	1.2%

Summary

- ◆ SQL is very useful for many aspects of data mining
 - Testing hypothesis
 - Calculating key metrics
 - Creating customer signatures
- ◆ SQL is a powerful tool
 - Takes advantage of parallel hardware and software (ahead of traditional analysis tools)
- ◆ When analyzing data for business value, we do not have to be proud (or ashamed) of the tools we use
 - The results are what's important
- ◆ My contact: gordon@data-miners.com



Thank you..



- First 20 to register and attend today will receive a prize
 - Gordon's "Data Analysis using SQL and Excel" book
- Emails will be sent to the winners on Monday Feb 1st.
- Books should arrive in a few weeks.
- Webinar is Recorded and can be found at www.xtremedata.com/accelerationacademy
- Please fill out the feedback forms – We read each and everyone of them.



Analytics. The wait is over.